Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#1

Version Control Systems

Autor:
Scripnic Alexandru

lector asistent:
Irina Cojanu
lector superior:
Svetlana Cojocaru

Laboratory work #2

1 Scopul lucrarii de laborator

Studierea unui Version Control Systems și lucrul cu versiunea CLI a acestuia. Examinarea instrucțiunilor de bază și lucru cu un repositoriu remote.

2 Obiective

– Intelegerea si folosirea CLI (basic level)

– Administrarea remote a masinilor linux machine folosind SSH (remote code editing)

– Version Control Systems (git —— bitbucket —— mercurial —— svn)

3 Laboratory work implementation

3.1 Tasks and Points

– Aplicatia trebuie sa fie dezvoltata si testata cu orice SDK ce include Emulator

– Testarea aplicatiei pe un device real

– Aplicatia trebuie sa suporte device-urile cu diferite rezolutii

– O alta aplicatie sofisticata la alegere: Game

– Foloseste libraria cross platform pentru a realiza o apliacatie cross platform (aplicatia poate fi compilata atit pe Android, cit si pe iOS)

3.2 Analiza lucrarii de laborator

Repository link

Primul pas spre elaborarea jocului a fost studiere api-ului de bază a IDE-ului Unity și lucrul cu scena de bază a lui [1]. Pentru început am studiat GameObjects și metoda de a manipula Componentele lui de gen Transform, Rigidbody2D, BoxCollider2D, etc.

Al doilea pas a fost deciderea jocului și tema lui. După navigarea mai detaliată pe GooglePlay am decis să creez un Endless Runner, care este mai simplu pentru un începător și e mai ușor la generarea nivelului.

Prima problemă întâlnită în joc a fost metoda de miscare a nivelului și mai precis:

a) Personajul va sta pe loc, iar restul nivelului se va mișca, iar personajul principal va sta pe loc.

b) Personajul va sta pe loc, iar restul nivelului nu se va mișca

Ambele au si pro și con, ca exemplu varianta a ne dă voie să nu simulăm aplicarea forțelor asupra personajului și con că trebuie să mișcăm tot nivelul și inclusiv obiectele generate dynamic. După studirerea a mai multor tutoriale[2], am decis să mă țin de varianta (b).

Al treilea pas a fost de a crea obiecte de bază, Sprites, care seminificau Player, Obstacles, Ground. Primul script[3] creat a fost pentru Player care aplică o viteză de mișcare asupra obiectului(Rigidbody2D) ce îl reprezintă pentru a se misca pe elementul Ground. Apoi am creat un punct invizibil care prin intermediul unui script determină daca trebuie mutată copia elementul[4] Ground în fața jucătorului pentru a crea o iluzie că pământul nu este finit.

Al patrulea pas a constituit în crearea a scriptului ce genera dinamic obstacolele de care personajul ar trebui să se ferească.

După finisarea mecanicii jocului, jucătorul avea posibilitatea de a face 3 acțiuni:

– Run - Stare de bază a jucătorului este de a fugi

– Jump - Personajul sare peste obstacole

– Slide - Personajul se lușuie sub obstacole

Următorul pas constă în adăugarea texturilor și animațiilor necesare pentru finisarea jocului. Pentru jucător am implementat 1 textură și 4 animații[5]: Run, Jump, Slide, Lose

- Run 3.1
- Jump 3.2
- Slide 3.3
- Lose 3.4

În urma animației Slide am fost nevoit să editez dynamic zona de impact cu alte obiecte (BoxCollider2D) asa cum poziția lui se schimbă.

Pentru obstacole a fost mai greu, deoarece am fost nevoit sa le editez manual pe fiecare în parte și modificare cu obiectul original, care era doar un dreptunghi, și editarea zonei de impact cu utilizatorul (BoxCollider2D). Texturile pentru Ground a fost mai ușor de găsit așa cum este mai des folosite în diferite jocuri, dar versiunea current de Unity nu permite repetarea unui texture pe zona lui de mai multe ori, și am fost nevoit să creez o textură de dimensiunea exactă la Ground.

Ultemele detalii în crearea jocului a fost adăugarea scorului și testarea aplicației pe un dispozitiv mobil.

3.3 Imagini

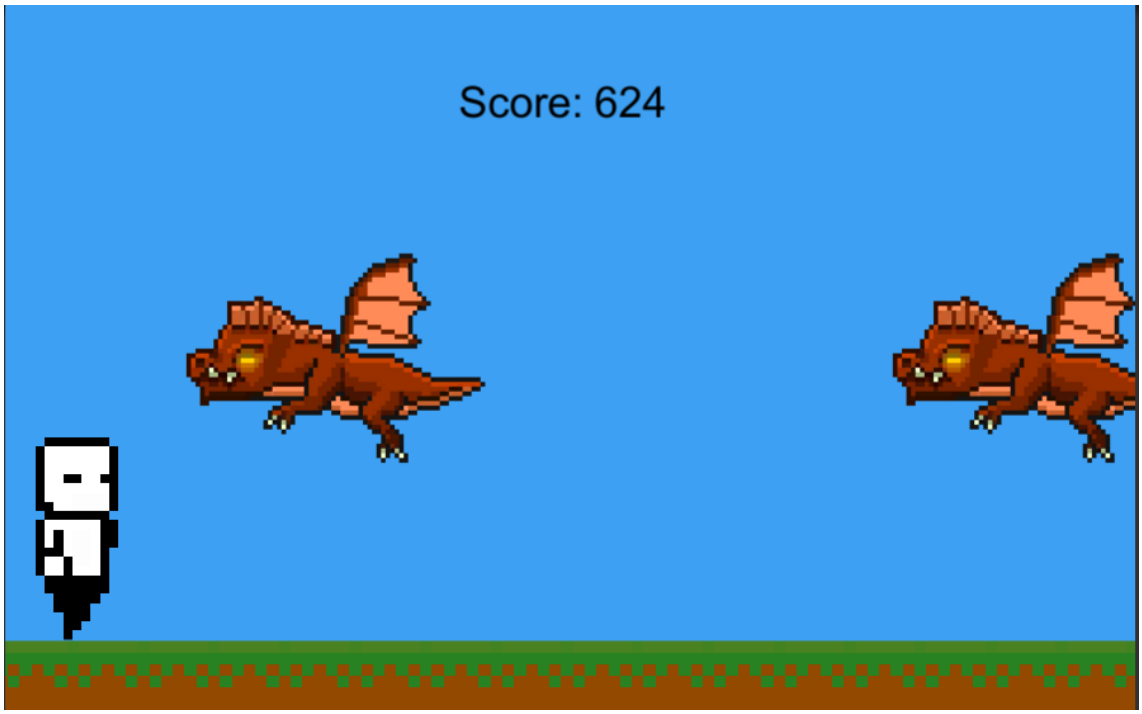

Figure 3.1– Running

Figure 3.2– Jumping

Figure 3.3– Sliding

Figure 3.4 – lose

Concluzie

Pe parcursul elaborării lucrării date am studiat un IDE nou Unity, care permite lansare aplicațiilor pe o listă mare de platforme, ca ex: Android, Windows, iOS, OSx, Playstation etc., ce oferă un API foarte user friendly pentru developerii care nu au mai lucrat întru-un asemenea domeniu.

Am profitat de avantajele sistemului de control a versiunilor, git, în momentul când am șters content de care aveam nevoie.

Am avut posibilitatea de a învăța diferențele dintr-un limbaj tipizat, C#, și unul netipizat, UnityScript, și avantajele și dezavantajele fiecărui în parte.

Am învățat care sunt bazele unui joc și care sunt pașii obligatorii care trebuie executați pe parcursul creării unui joc, din greșelile admise de mine:

- Căutarea texturilor se face înainte de a începe lucrul asupra jocului
- Căutarea texturilor se face pentru tot jocul nu doar pentru personaj (în cazul în care nu ai designer skills)
- Dimensiunile obiectelor se fac pe bază de forma care o va avea după aplicarea texturilor
- Se studiază limbajele de programarea disponibile pentru un IDE apoi se începe lucrul, asa cum unul poate avea avantaje mai mare în cadrul unui IDE decât altele

References

1 Unity, *official page*, https://unity3d.com/learn/tutorials/topics/2d-game-creation

2 Unity Guide, *official page*, https://unity3d.com/learn/tutorials/topics/scripting/lets-make-game-infinite-runner

3 Unity Scripts, *official page*, https://unity3d.com/learn/tutorials/topics/scriptingr

4 Unity Instantiate, *official page*, https://docs.unity3d.com/ScriptReference/Object.Instantiate.html

5 Unity Animation, *official page*, https://unity3d.com/learn/tutorials/topics/animation

6 Transform, *official page*, https://docs.unity3d.com/ScriptReference/Transform.html

7 Rigid Body, *official page*, https://docs.unity3d.com/ScriptReference/Rigidbody2D.html

8 Box collider, *official page*, https://docs.unity3d.com/ScriptReference/BoxCollider2D.html